Scalable Event-based Clustering of Social Media via Record Linkage Techniques

Timo Reuter, Philipp Cimiano
Semantic Computing
CITEC, University of Bielefeld
Bielefeld, Germany
{treuter,cimiano}@cit-ec.uni-bielefeld.de

Lucas Drumond, Krisztian Buza, Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
University of Hildesheim
Hildesheim, Germany
{ldrumond,buza,schmidt-thieme}@ismll.de

Abstract

We tackle the problem of grouping content available in social media applications such as Flickr, Youtube, Panoramino etc. into clusters of documents describing the same event. This task has been referred to as event identification before. We present a new formalization of the event identification task as a record linkage problem and show that this formulation leads to a principled and highly efficient solution to the problem. We present results on two datasets derived from Flickr – *last.fm* and *upcoming* – comparing the results in terms of Normalized Mutual Information and F-Measure with respect to several baselines, showing that a record linkage approach outperforms all baselines as well as a state-of-the-art system. We demonstrate that our approach can scale to large amounts of data, reducing the processing time considerably compared to a state-of-the-art approach. The scalability is achieved by applying an appropriate blocking strategy and relying on a Single Linkage clustering algorithm which avoids the exhaustive computation of pairwise similarities.

1 Introduction

As the amount of data uploaded to social media portals such as Flickr, Youtube, Panoramino, etc. keeps proliferating, techniques for structuring this massive content become crucial to better organize and manage this information. One obvious candidate for organizing the content according to topics are clustering approaches. Clustering approaches have been recently applied to social media data, for example for grouping Flickr data into clusters of pictures describing the same event. This task has been dubbed *event identification* (Becker, Naaman, and Gravano 2010).

There are two important challenges in applying clustering algorithms to social media data. First, clustering techniques need to scale to volumes of data behind such social media portals. Flickr for example features 5.5 billion images as of January 31, 2011 and around 3000-5000 images are added per minute. It is thus an important research challenge to develop clustering approaches that can handle such large datasets. Second, it is crucial to produce clusters that are meaningful to users, in particular to produce clusters at the level of granularity that users want. One way of approximating the form of clusters that users want to see is to apply supervised clustering approaches which learn to approximate the clusters from user-labeled data. Becker et al. (2010) for example use labeled data extracted from *last.fm*[1] and *upcoming*[2] to tune the similarity measure and cluster assignment threshold to match the classes of events specified by users. This is indeed an attractive way of learning the type and form of clusters that users wish to see.

In this paper we show that event identification, i.e. the task of identifying documents describing the same event, can be naturally phrased as a record linkage task. Record linkage (also called *duplicate detection* or *object identification*) is the task of identifying different descriptions of the same real-world entity (Fellegi and Sunter 1969). Recently, record-linkage methods have been applied to various tasks such as discovering references of the same product over different databases, recognition of citations referring to the same scientific publication (McCallum, Nigam, and Ungar 2000), and identification of web-pages describing the same person (Romano et al. 2009).

We present a novel adaptive and scalable approach to organize social media data into event-based clusters which is based on a state-of-the-art record linkage algorithm (Rendle and Schmidt-Thieme 2006). The approach is adaptive in the sense that it uses machine learning techniques to learn an optimal similarity measure on the basis of training data (see Becker, Naaman, and Gravano 2010) and is efficient as it avoids computing the similarity between all data pairs. Thus, it addresses both of the challenges mentioned above. In fact, we apply our approach to two datasets derived from Flickr which consist of 349,996 and 1,492,883 pictures, respectively. We refer to these datasets as *last.fm* and *upcoming*, as the labels have been extracted from these sites. When clustering datasets of the scale we consider, i.e. datasets consisting of hundred of thousands or even millions of items, devising efficient approaches is a major issue as the cost of calculating all the pairwise similarities – as required by most clustering approaches – is prohibitive. One strategy for achieving scalability is *blocking*, which reduces the number of pairs of documents considered and thus allows the ap-

[1] http://www.lastfm.de/events
[2] http://upcoming.yahoo.com/

proach to scale-up.

In this paper, we provide the following contributions:

1. We give a principled formulation of event identification as a record linkage task. Formulating event identification in social media as a record linkage task has a number of benefits: it represents a principled formulation of the problem allowing us to apply the whole body of techniques developed in the field of record linkage. Further, it allows to scale up the task of event identification by using standard techniques applied to record linkage, such as blocking (Michelson and Knoblock 2006; Baxter, Christen, and Churches 2003).
2. We show that using Single Linkage clustering as global decision model allows us to scale to the large datasets we consider by avoiding the computation of all pairwise similarities, an approach that is faster than the one in Becker, Naaman, and Gravano (2010). In fact we apply a variant of Single Linkage clustering for the record linkage task that scales to the amounts of data we consider by i) circumventing the need of recalculating the cluster-similarities after a merge and ii) being able to operate with a sample of the pairwise similarities only. We show that in combination with blocking we yield an approach which outperforms the state-of-the-art approach of Becker et al. (2010) while reducing the time needed to process the data considerably.
3. We also evaluate our approach in a transfer learning mode (Raina et al. 2007) and show that the parameters learned on one dataset (*upcoming*) transfer well to the other dataset we consider (*last.fm*) without major performance drop with respect to cluster quality (F-measure decreases by 5.3 percentage points and NMI by 0.1). This shows that the approach could effectively avoid overfitting while learning the clusters that appear in one dataset and therefore the model could be applied to another similar dataset.

The paper is structured as follows: in Section 2 we describe how the event identification problem can be formulated as a record linkage problem and present our approach. In Section 3 we describe our data, document representation, similarity measures, baselines and present our results. We discuss related work in Section 4 and conclude in Section 5.

2 Event Identification as a Record Linkage Task

The task of Record Linkage consists in inducing an equivalence relation over a set of objects S. We assume there is a set of objects $S_Y \subset S$ for which the equivalence relation is known a priori. The task is to group the rest of the objects $S_X = S \setminus S_Y$ according to the equivalence relation (Rendle and Schmidt-Thieme 2006).

Event identification can be defined as a record linkage task: given a set of social media documents S where each document is associated to an unknown event, we aim at discovering which documents are associated to the same event. In this case, the equivalence relation is the SameEvent $\subseteq S^2$ relation. For a set of documents S_Y the equivalence relation is known. The rest of the documents $S_X = S \setminus S_Y$ has to be grouped according to the equivalence relation.

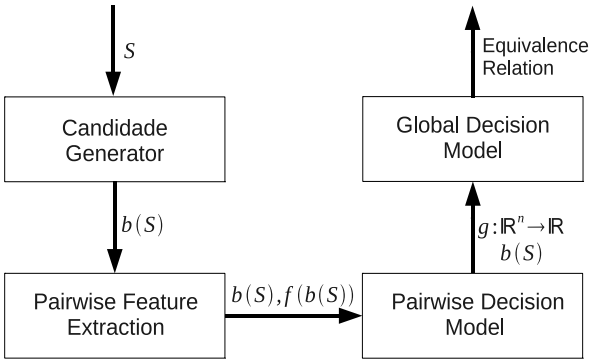

Figure 1: Record Linkage model based on the components identified by Rendle and Schmidt-Thieme (2006)

Our approach, depicted in Fig. 1, builds on a pipeline of the four major components that state-of-the-art record-linkage approaches typically rely on. Our approach works by comparing document pairs. Since the number of pairs is $O(|S|^2)$, it is infeasible to consider all possible pairs (especially for the large datasets we consider). Thus a *candidate pair generator* or *blocker* is applied to select a subset of the set of all possible pairs to be considered, such that pairs of obviously different objects are discarded, thus allowing to scale to large datasets (Section 2.1). Pairwise feature extraction can be seen as a function $sim : S^2 \rightarrow \mathbb{R}^n$ that takes a pair of objects/documents as input and creates a real-valued feature vector describing their similarity along different dimensions, as described in Section 2.2. These features are used to determine whether the equivalence relation holds for the given pair. This is done by the pairwise decision model described in Section 2.3. The output of the pairwise decision model is the likelihood that both documents (s_1, s_2) belong to the same equivalence class. In our case this represents the likelihood that documents s_1 and s_2 describe the same event, i.e. the likelihood that $(s_1, s_2) \in$ SameEvent. This output can be viewed as a function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ that maps the pairwise feature vector of a document pair to the likelihood that both documents refer to the same event. As pairwise decision model, a regression model is used that outputs continues values for $g(sim(s_1, s_2))$. Therefore, we can use the output of the pairwise decision model, $g(sim(s_1, s_2))$ as a similarity measure in the *Global Decision Model* that produces a complete clustering. As global model we use Single Linkage (Duda, Hart, and Stork 2001).

2.1 Blocker

In order to make the approach scalable, only a subset of the document pairs is considered. This subset is selected by a *blocker*. Previous work (McCallum, Nigam, and Ungar 2000; Rendle and Schmidt-Thieme 2008b) has indeed shown that an appropriate blocker is able to speed up the approach while not substantially decreasing the quality.

Formally, a blocker can be seen as a function that selects a subset of the set of all pairs of documents: $b : \mathcal{P}(S) \rightarrow$

$\mathcal{P}(S^2)$, with $b(S) \subseteq S^2$. If $(s_1, s_2) \in b(S)$, then this pair will be considered later on; otherwise the pair is eliminated.

In order to build a blocker for the event identification task, one can take advantage of the fact that events typically extend over a limited time interval. Thus, multimedia documents that largely differ on their creation time are much less likely to belong to the same event. We apply a moving window strategy in order to generate a set of candidate pairs. First, the documents are ordered according to their creation time. Then, for each document s, a window W_s^n containing the next n documents is created. For each document $s' \in W_s^n$, the time similarity (defined in Section 2.2) between s and s' is computed and if it is higher than a threshold Θ_t then the pair (s, s') is returned as a candidate pair. Both n and Θ_t are hyperparameters. Let S^O be the ordered set of documents in S sorted according to their creation time and s_i^O the i-th document according to this ordering. The blocker $b(S)$ can be defined as:

$$b(S) := \{(s_i^O, s_j^O) | 0 < j - i < n \wedge sim_{time}(s_i^O, s_j^O) > \Theta_t\}$$

Thus, we consider a number of at most $|S| \times n$ pairs for which the similarity needs to be computed. The hyperparameters are optimized using grid search on the training set. The results of this search in the experiments conducted here are presented in Section 3.1.

2.2 Pairwise Feature Extraction

In this section we describe the specific features that are computed on the basis of the metadata described above to represent a picture. We also describe the similarity measures that are used to compare two pictures along each of these feature dimensions. We use the same features and similarity measures as Becker et al. (2010):

- **date/time** we define the similarity between two date/time values of the documents d_1 and d_2 as $sim_{time}(d_1, d_2) = 1 - \frac{|t_1 - t_2|}{y}$ where t_1 and t_2 are date/time values represented as the number of minutes elapsed since the Unix epoch and y is the number of minutes of one year. For the specific case that t_1 and t_2 are more than one year apart, we set their similarity to 0.
- **geographic feature**: the location of a picture is described in terms of latitude and longitude coordinates. We use the haversine formula to determine the great-circle distance between two points on earth from these coordinates. The similarity is thus defined as $sim_{geo}(d_1, d_2) = 1 - H(L_1, L_2)$ where

$$H(L_1, L_2) = 2 \cdot atan^2(\sqrt{\phi}, \sqrt{1-\phi})$$
$$\phi = sin^2(\frac{\Delta lat}{2}) + cos(lat_1) \cdot cos(lat_2) \cdot sin^2(\frac{\Delta lon}{2})$$
$$\Delta lat = lat_2 - lat_1, \Delta lon = lon_2 - lon_1$$

- **textual features**: including tags, title and description are described via TF-IDF vectors. The cosine of the angle between two vectors is used as similarity measure. We apply the Porter stemmer in order to normalize tags following the advice of Becker et al. who observe that their results improved when using stemming:

$$sim_{text}(d_1, d_2) = \frac{\sum_t tfidf(t, d_1) \cdot tfidf(t, d_2)}{\sqrt{\sum_t tfidf(t, d_1)^2} \cdot \sqrt{\sum_t tfidf(t, d_2)^2}},$$

where t is a token and d_1, d_2 are documents.

Using these features, a feature vector for a pair of documents d_1 and d_2 looks as follows:

$$sim(d_1, d_2) = \begin{pmatrix} sim_{time}(d_1, d_2) \\ sim_{geo}(d_1, d_2) \\ sim_{tags}(d_1, d_2) \\ sim_{title}(d_1, d_2) \\ sim_{description}(d_1, d_2) \\ sim_{notes}(d_1, d_2) \\ sim_{WOEID}(d_1, d_2) \\ sim_{alltext}(d_1, d_2) \end{pmatrix}$$

The feature sim_{text} is applied to all textual features like sim_{tags}, sim_{title}, etc.

The metadata extracted includes: i) the date/time of capture and upload, ii) geographic position, iii) tags, iv) title, v) description, vi) notes inside the picture, vii) WOEID[3], and viii) the owner. We extend both datasets by a feature called "Alltext". Alltext combines all textual features in one: tags, titles, descriptions and WOEID as a flattened text representation. The rationale for this is that, given two pictures representing the same event, a relevant term (e.g. the name of the place) could appear in the tags of one picture and in the title of the other. Considering tags and title as separate dimensions would thus underestimate the similarity of these pictures. Merging all textual features is a way to circumvent this problem.

2.3 Pairwise Decision Model

The goal of the pairwise decision model is to estimate the likelihood that the equivalence relation holds for a given pair of documents based on their pairwise features, i.e. the pairwise decision model is a function $g : \mathbb{R}^n \rightarrow \mathbb{R}$. In order to make local decisions about whether two documents belong to the same event, any model producing a continuous output (e.g. SVM-regression, Nearest Neighbor Regression, Linear Regression, etc.) can be used. While training the model, the target is binary, in particular 1 and 0 depending on whether both objects s_1 and s_2 belong to the same equivalence class or not, i.e. if $(s_1, s_2) \in$ *SameEvent* or $(s_1, s_2) \notin$ *SameEvent*.

We use SVMs since they have been shown to perform well on the task of identifying events (see Becker, Naaman, and Gravano 2010). Further, SVMs have also been used in state-of-the-art record linkage approaches (Bilenko and Mooney 2003; Rendle and Schmidt-Thieme 2008a). In this paper we use the C-SVC model from the libSVM package (Chang and Lin 2001). In order to produce continuous output, we used its probability estimates feature.

As stated in the beginning of this section, we assume that equivalence relation is known for a subset of objects

[3] Where-On-Earth ID

$S_Y \subset S$. Thus, the pairwise decision model is trained using pairs from S_Y^2 selected by the blocker, i.e. $b(S_Y) \subset S_Y^2$. The problem with this approach is that it generates an unbalanced training set. Thus a set $S_Y^{Train} \subseteq b(S_Y)$ is sampled by randomly selecting from $b(S_Y)$ a fixed number of positive pairs (pairs for which the $SameEvent$ relation holds) and negative pairs (pairs for which the equivalence relation does not hold). In our experiments we determined (using only the training set) that choosing 5000 positive and 5000 negative pairs was enough for getting an effective pairwise decision model. This result is in line with Becker, Naaman, and Gravano (2010). The training data is composed of the features of the pairs S_Y^{Train} and their respective labels.

2.4 Global Decision Model (SL algorithm)

As a global decision model, we rely on Single Linkage clustering (SL) (Duda, Hart, and Stork 2001).

Due to the huge volume of data, our variant only processes the list of pairs produced by the blocker, being linear in the number of pairs – in our case at most $|S| \times n$ where n is the size of the window used by the blocker. Alg. 1 depicts this variant of SL in pseudocode. Note that, when two clusters are merged, the similarity between pairs of documents belonging to the new cluster is no longer relevant, such that not all the pairs need to be evaluated by the algorithm. [4]

Algorithmus 1 SL(S_X,$b(S_X)$,$\Theta_{cluster}$)

$P \leftarrow \{\{s\}|s \in S_X\}$ (P stands for "Partitioning")
$Pairs \leftarrow \{(s_i, s_j)|s_i, s_j \in b(S_X) \wedge g(sim(s_i, s_j)) \geq \Theta_{cluster}\}$
for all $(s_i, s_j) \in Pairs$ **do**
  $c_i \leftarrow \{c_i|c_i \in P \wedge s_i \in c_i\}$
  $c_j \leftarrow \{c_j|c_j \in P \wedge s_j \in c_j\}$
  if $c_i \neq c_j$ **then** $P \leftarrow (P \backslash \{c_i, c_j\}) \cup \{c_i \cup c_j\}$
end for

return P

3 Experimental Setup and Results

In this section we present our experimental settings and results. In particular we describe how the *last.fm* and *upcoming* datasets have been derived from Flickr. We evaluate all approaches on the two datasets using Normalized Mutual Information (NMI) and F_1-measure (F).

3.1 Experimental Settings

Datasets For our experiments we use two different datasets which we refer to as *last.fm* and *upcoming*, respectively. While *last.fm* includes only events in the area of music such as concerts, *upcoming* provides all different kind of events which are of public interest. We consider only pictures for our experiments assigned to a specific event via a machine tag. Such machine tags represent unique event IDs – such as "lastfm:event=679065" or "upcoming:event=23066" originating from *last.fm* and yahoo.upcoming.com. The Flickr pictures on `http://www.flickr.com/photos/tags/upcoming:event=237347` for example belong to the event `http://upcoming.yahoo.com/event/237347` in *upcoming*. This assignment of pictures to events via machine tags can be used to construct a gold standard for clustering by assuming that all pictures with the same machine tag belong to the same event.

Table 1: The datasets

Dataset	Number of pictures	Number of events
last.fm	1,492,883	58,366
upcoming	349,996	12,061

We downloaded Flickr pictures with machine tags having a lastfm:event or upcoming:event as prefix using the Flickr API. The *last.fm* dataset contains 1,492,883 pictures, spread over 58,366 events. The second dataset – *upcoming* – consists of 349,996 pictures spread over 12,061 unique events. While all pictures contain metadata for the time of capture and upload as well as author, not all pictures include tags or a title. Only 36% have geographic information assigned and 34% a description. In both datasets the pictures are not equally distributed. In the *upcoming* dataset, a cluster contains 29.02 pictures on average. The *last.fm* dataset has an average cluster size of 25.58 pictures.

In order to approximate real settings, we remove the machine tags from the data and use them only to create the gold standard.

If a similarity can not be determined due to missing metadata (e.g. a picture does not include location data), a similarity of 0 is assumed.

Evaluation Measures In order to evaluate the performance of all algorithms, we use standard evaluation measures from the text mining literature typically used in clustering tasks: Normalized Mutual Information (NMI) and F_1-Measure.

NMI has been used as an objective function for cluster ensembles (Strehl and Ghosh 2003). NMI quantifies the amount of information which is shared between two clustering results. Specifically, NMI is defined as follows:

$$NMI(C,E) = \frac{2 \cdot \sum_k \sum_j \frac{|e_k \cap c_j|}{n} \cdot log \frac{n \cdot |e_k \cap c_j|}{|e_k| \cdot |c_j|}}{-\sum_j \frac{|c_j|}{n} \cdot log \frac{|c_j|}{n} + -\sum_k \frac{|e_k|}{n} \cdot log \frac{|e_k|}{n}},$$

where $C = c_1, ..., c_j$ and $E = e_1, ...e_j$ are two sets of clusters which are compared. Each c_j and e_k is a set of documents and n is the total amount of documents.

Further, we also use the well-known F_1-Measure as the harmonic mean of precision and recall. Precision and Recall are computed and averaged document-wise as in Becker et al. (2010):

[4] Note that SL can be considered as a variant of Hierarchical Agglomerative Clustering (Duda, Hart, and Stork 2001). For our work, however, the hierarchy of clusters is not relevant, such that we build on the simplified algorithm shown in Alg.1.

$$P_b = \sum_{d \in D} \frac{1}{|D|} \frac{|Cluster(d) \cap GoldStandard(d)|}{|Cluster(d)|}$$

$$R_b = \sum_{d \in D} \frac{1}{|D|} \frac{|Cluster(d) \cap GoldStandard(d)|}{|GoldStandard(d)|}$$

$$F_1\text{-}Measure = 2 \cdot \frac{P_b \cdot R_b}{P_b + R_b},$$

Data Splits In order to stay as close as possible to a realistic application scenario, we order the documents in the datasets by their time of upload. In a real-world scenario, new documents also arrive in exactly this temporal order. We then divide both datasets into three equal parts. The first and second part are used as training and validation set, respectively. For the first part, the machine tags are used as a ground truth for training the pairwise decision model as discussed in Section 2.3. The third is reserved for evaluation purposes. The models are trained on the training set while hyperparameters are tuned using the validation set. The hyperparameters in the case of the record linkage approach are the complexity constant C of the SVM and the exponent e of the RBF kernel of the SVMs, the threshold parameter Θ_t and the window size n for the blocker as well as the $\Theta_{cluster}$ threshold used in the SL algorithm.

Baselines In this section we present the quantitative results of our approach (RL-SL) both with respect to performance in terms of precision, recall, F-Measure and NMI as well as efficiency (measured in terms of runtime). In particular, we compare the results of our approach (RL-SL) to seven baselines:

- **TimeDay**: all pictures taken on the same day are placed together into one cluster.
- **Geo**: all pictures in a window of 0.009° (∼1.0 km) and 0.011° (∼1.2 km) for the latitude and longitude, respectively, are placed into one cluster. These values are the average distance of pictures belonging to the same event in the gold standard (all events that were further away than 1° were eliminated to reduce noise).
- **TimeGeo**: this baseline is a combination of TimeDay and Geo. Here the clusters of TimeDay are subdivided using the windows of the Geo baseline.
- **Owner**: all pictures of one uploader are put into one cluster.
- **Tags**: the pictures are clustered using only the cosine between TF-IDF vectors as similarity measure.
- **IC-Ens, Incremental Clustering (Ensemble)**: Becker et al. propose the usage of a single-pass incremental clustering algorithm with a threshold parameter (Becker, Naaman, and Gravano 2010). The algorithm loops over all documents (in one pass) and assigns each document to its closest cluster, provided this similarity is over a specified threshold, creating a new cluster otherwise. The overall similarity of two documents is the linear combination of the single similarities. The ensemble represents a weighted linear combination of all the similarity measures whereby the weights are optimized independently of each other. See (Becker, Naaman, and Gravano 2010) for details.
- **IC-SVM, Incremental Clustering (SVM)**: Instead of an ensemble, a binary SVM is used to determine whether two documents belong together or not.

Optimization of parameters In IC-Ens, the overall similarity between a document and a cluster is assumed to be a linear combination of the single similarities. The weight and threshold hyperparameters for the ensemble are optimized on training data separately for each similarity via exhaustive search as described in Becker et al. (2010). These individual similarities are the following: $sim_{time}(d_1, d_2)$, $sim_{geo}(d_1, d_2)$, $sim_{tags}(d_1, d_2)$, $sim_{title}(d_1, d_2)$, $sim_{description}(d_1, d_2)$, $sim_{notes}(d_1, d_2)$, $sim_{WOEID}(d_1, d_2)$, $sim_{alltext}(d_1, d_2)$. In the IC-SVM and RL-SL approaches, a binary SVM is trained to learn a linear model over these features. Correct data point-cluster assignments are used as positive examples, incorrect assignments are used as negative examples. The hyperparameters for RL-SL as well as IC-SVM were optimized by systematic search and evaluation of parameters on the validation set. Using an RBF kernel we determined C = 8 and e = 2 as optimal parameters. Using the same methodology, the RL-SL hyperparameters $\Theta_{cluster}$, Θ_t and n were set to 0.5, 0.96 and 700 respectively.

3.2 Results

Table 2 shows the results in terms of NMI and F-Measure on the *upcoming* test set. The results in table 2 license the conclusion that our novel record-linkage based approach (RL-SL) achieves the best results compared to all other approaches. In particular, it clearly outperforms all naive baselines relying on a single source of evidence (TimeDay, Geo, Owner, Tags) as well as the simple combinations of the time and geo features (TimeGeo). It further outperforms the best naive baseline using only tags as evidence by 12.0 percentage points and the state-of-the-art approach of Becker et al. by 9.6 percentage points regarding the F-Measure. Although RL-SL relies on the same pairwise decision model as IC-SVM (Becker, Naaman, and Gravano 2010), i.e. an SVM trained on the same set of features, RL-SL makes use of a different clustering algorithm and a blocking strategy which is better suited for the event identification scenario. The SL algorithm only evaluates the relevant pairs of documents returned by the blocker in contrast to IC-SVM which compares each document with the centroid of each cluster. Having shown that our approach outperforms all other naive baselines as well as the state-of-the-art incremental clustering approach of Becker et al. on the *upcoming* dataset, we now turn to two important questions. First, we analyze in how far the clustering model learned on one dataset (*upcoming*) can be transferred to a similar dataset (*last.fm*) without retraining on the new dataset. This setting is typically referred to as *transfer learning* in the machine learning community. Second, we also compare the processing time for the different approaches, showing that our approach outper-

Table 2: Quality of the classification algorithms and the baselines over the *upcoming* test set

Algorithm	NMI	F_1-Measure
TimeDay	84.7%	49.1%
Geo	53.0%	38.4%
TimeGeo	89.1%	64.0%
Owner	83.9%	52.8%
Tags	91.3%	73.0%
IC-Ens	88.6%	62.8%
IC-SVM	92.5%	75.4%
RL-SL	**95.9%**	**85.0%**

Table 3: Quality of the classification algorithms and the baselines over the *last.fm* test set

Algorithm	NMI	F_1-Measure
IC-SVM	88.2%	73.0%
RL-SL	95.8%	79.7%

forms the incremental clustering approach considerably. Actually, an important feature of our approach is that the parameter n which determines the number of document pairs that are fed into the Single Linkage clustering algorithm, allows to trade-off quality for performance or the other way round. By increasing the size of the window, we increase the quality at the cost of a higher runtime, as more pairs need to be processed (see Fig. 2). This feature of our approach allows it to scale to data of arbitrary size as long as one is willing to put up with a corresponding loss in quality.

Transfer In the transfer learning (Raina et al. 2007) experiment, we simulate the situation where no labeled data is available in a domain (e.g. because the domain is new or annotation costs are high etc.) but we have access to labeled data in a similar domain and the model is to be trained using that data. The result of applying the clustering approaches (IC-SVM and RL-SL) with the parameters obtained from the training data of the *upcoming* dataset on the *last.fm* dataset are shown in table 3. It is remarkable that both approaches have a very good performance on this dataset given the fact that they have not been tuned to the dataset. One explanation for this is that the clusters in the *upcoming* and *last.fm* data are very similar. Nevertheless it is worth emphasizing that the decrease in performance is more severe for IC-SVM. This shows that our approach can be better transferred to new data without retraining the parameters.

Efficiency Considerations As mentioned in the introduction, one of the important challenge in clustering social media data lies in developing approaches that can scale to the massive volumes of data generated in social media portals such as Flickr. In this section we thus discuss and compare the runtime of the most successful approaches: IC-SVM and RL-SL. In addition, we also compare the number of similarity computations needed, which directly correspond to the number of predictions that the SVM has to perform. Table 5 shows the processing time for the different methods. Our experiments have been carried out on a standard

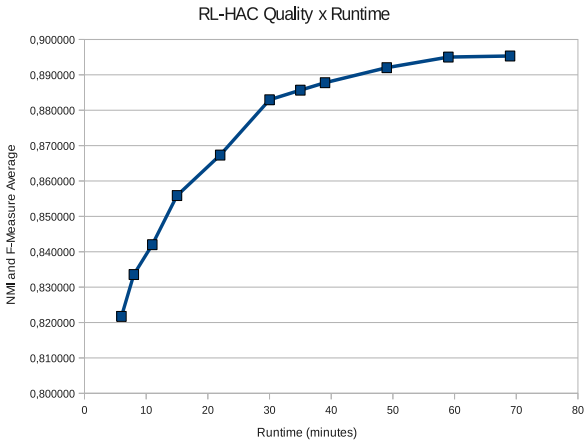

Figure 2: Performance of RL-SL on the *Upcoming* validation set for a window size raging from $n = 50$ to $n = 700$. On the horizontal axis there is the runtime of the approach and on the vertical axis, the average between NMI and F-Measure.

Table 4: Number of SVM predictions (in Millions) for RL-SL ($n = 700$) and IC-SVM on both datasets

	upcoming	last.fm
RL-SL	70	264
IC-SVM	231	3987

Intel Xeon 2.5GHz machine with 8GB memory. The time unit used in the table corresponds to 4103s (the time needed by our RL-SL-approach to process the complete *upcoming* dataset). The results can be summarized as follows:

- Our RL-SL approach is more efficient compared to the IC-SVM approach. Using the hyperparameters that deliver the best prediction quality, the IC-SVM approach takes 2.6 times longer to cluster the *upcoming* test set and 12 times longer to cluster the *last.fm* dataset.
- There is a trade-off between the effectiveness and runtime of the RL-SL approach, controlled by the blocker hyperparameter n (the window size). Larger window sizes incur in larger runtimes, as more pairs are returned by the blocker, but also in better quality. This behavior can be seen in Fig. 2 where the runtime and effectiveness are plotted for various window sizes.

The much higher runtime increase of IC-SVM compared to the nearly linear growth of RL-SL is due to the fact that the incremental clustering approach compares every document with every single cluster built so far to determine which cluster to assign the document to. This exhaustive search is very expensive and leads to a high number of SVM predictions. The number of SVM predictions for both approaches and both datasets are shown in table 4. While RL-SL requires 70 Mio. similarity computations on the *upcoming* dataset, IC-SVM requires 3.3 times more similarity computations. The difference on the *last.fm* dataset is a factor of 15.

Table 5: Runtimes for one clustering (in time units)

	Test Set (upcoming)	Test Set (last.fm)
IC-SVM	2.6 (3h)	46.0 (52.5h)
RL-SL	1.0 (1.14h)	3.8 (4.4h)

In table 6 the absolute runtimes for the RL-SL algorithm are given, both on the training and test datasets and broken down for each step.

4 Related Work

In this paper we cast the task of clustering Flickr data into clusters of images representing the same event as a record linkage problem. Record linkage (also called object identification and duplicate detection) aims at identifying which references (social media documents in our case) belong to the same real-world object (an event respectively). State-of-the-art record linkage approaches rely on models learned from partially labeled data using machine learning techniques (Christen 2008). One prominent application area of record linkage techniques is coreference resolution (Soon, Ng, and Lim 2001; Ng and Cardie 2002), the task of determining whether two natural language expressions are references to the same real world entity. Usually, the core of a machine learned record linkage system is a pairwise decision model that predicts the likelihood of instances being equivalent: Cohen and Richmann (2002) and Sarawagi and Bhamidipaty (Sarawagi and Bhamidipaty 2002) used probabilistic classifiers, Singla and Domingos (2005) applied conditional random fields, whereas Rendle and Schmidt-Thieme (2006) built on SVMs, while Buza et al. (2011) used linear regression and multilayer perceptrons. In a subsequent step, called *collective or global decision*, the pairwise decisions are integrated and a consistent prediction is created. A simple solution for this subproblem is to take the transitive closure over the predicted equivalent pairs, like Singla and Domingos (2005) and Bilenko and Mooney (2003) propose. More sophisticated methods cluster instances based on their pairwise equivalence likelihoods (Cohen and Richman 2002; Rendle and Schmidt-Thieme 2006; Bilenko, Basu, and Sahami 2005).

In order to be able to apply record linkage for web data, usually scaling techniques are required: Canopy-blockers (McCallum, Nigam, and Ungar 2000) and adaptive blockers (Bilenko, Kamath, and Mooney 2006; Michelson and Knoblock 2006) are two of the most prominent scaling techniques. An overview of blocking techniques is given by Baxter et al (Baxter, Christen, and Churches 2003).

In the area of event identification in social media, there have been attempts to learn a classifier that learns to distinguish Flickr documents representing an event from those that do not (Rattenbury and Naaman 2009), e.g. Firan et al. (2010) used Naive Bayes classifiers. In contrast to our approach where clusters are not known a priori, the event classes are known beforehand in the approach of Firan et al. Becker et al. (2010) proposed an incremental clustering approach, similar to the one used for detecting events in text document streams (Allan, Papka, and Lavrenko 1998). In their approach, each document must be compared to all existing clusters before deciding in which cluster it will be added, and this can cause scalability issues. By using a more appropriate blocker and Single Linkage as global decision model, our approach reduces the number of comparisons to be made and also does not need to recompute cluster centroids everytime a new document is added to a cluster. In general, we are not aware of any work applying record linkage methods to the task of identifying clusters of images related to the same event, a task which has been referred to as *event identification* before (Becker, Naaman, and Gravano 2010).

5 Conclusions and Future Work

We have tackled the problem of finding clusters of pictures describing the same event in Flickr data, a task that has been dubbed *event identification* in previous work (Becker, Naaman, and Gravano 2010). We have formulated this task as a record linkage problem and presented a novel approach which relies on state-of-the-art techniques from the area of record linkage. In particular, the approach relies on an appropriate blocking strategy to reduce the number of pairs considered as well as on an efficient Single Linkage clustering algorithm with a threshold hyperparameter to cluster the documents. The advantage of this algorithm is that it does not require the exhaustive computation of pairwise similarities for all the documents to be clustered. Thus, our approach is scalable to large datasets consisting of millions of documents. We have shown on the one hand that our approach outperforms a state-of-the-art incremental clustering approach by Becker et al. (2010) in terms of Normalized Mutual Information and F-Measure on both datasets considered. On the other hand, we have also shown that the clustering model and parameters obtained on one dataset (*upcoming* in our case) can be successfully transferred to a similar dataset without significant performance degradation. This is important from a practical point of view as it shows that the model can be applied off-the-shelf without need of retraining. Most importantly, we have shown that our approach reduces the processing time considerably compared to the incremental clustering approach of Becker et al. (2010). Even further, we have shown empirically that the processing time seems to increase linearly with the size of the data for our approach, a very interesting property making our approach indeed scalable to much bigger datasets.

6 Acknowledgments

The research is funded by the Deutsche Forschungsgemeinschaft (DFG), Excellence Cluster 277 “Cognitive Interaction Technology”. Research partially supported by the Hungarian National Research Fund (Grant Number OTKA 100238). Lucas Drumond is supported by CNPq, a Brazilian Government institution for scientific development. The authors would like to thank Steffen Rendle for fruitful discussions.

References

Allan, J.; Papka, R.; and Lavrenko, V. 1998. On-line new event detection and tracking. In *Proceedings of the 21st An-*

Table 6: Runtimes for train and test (RL-SL)

	Training Set (upcoming)	Test Set (upcoming)	Test Set (last.fm)
Blocker	9s	89s	281s
Pairwise Features Extraction	24s	650s	2595s
Pairwise Decision Model (SVM)	3s	3308s	12705s
Global Decision Model (SL)	-	56s	207s
Total	36s	4103s	15788s

nual International ACM SIGIR Conference on Research and Development in Information Retrieval, 37–45.

Baxter, R.; Christen, P.; and Churches, T. 2003. A comparison of fast blocking methods for record linkage. In *Proceedings of the 2003 ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 25–27.

Becker, H.; Naaman, M.; and Gravano, L. 2010. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM International Conference on Web search and Data Mining*, 291–300.

Bilenko, M., and Mooney, R. J. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 39–48.

Bilenko, M.; Basu, S.; and Sahami, M. 2005. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the 5th IEEE International Conference on Data Mining*, 58–65.

Bilenko, M.; Kamath, B.; and Mooney, R. J. 2006. Adaptive blocking: Learning to scale up record linkage. In *Proc. of the 6th IEEE International Conf. on Data Mining*, 87–96.

Buza, K.; Nanopoulos, A.; and Schmidt-Thieme, L. 2011. Fusion of Similarity Measures for Time Series Classification. In *6th International Conference on Hybrid Artificial Intelligence Systems*.

Chang, C.-C., and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Christen, P. 2008. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proc. of the 14th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, 151–159.

Cohen, W. W., and Richman, J. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *Proc. of the 8th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, 475–480.

Duda, R.; Hart, P.; and Stork, D. 2001. *Pattern classification*. John Wiley & Sons, Inc.

Fellegi, I., and Sunter, A. 1969. A theory for record linkage. *Journal of the American Statistical Association* 64(328):1183–1210.

Firan, C. S.; Georgescu, M.; Nejdl, W.; and Paiu, R. 2010. Bringing order to your photos: event-driven classification of flickr images based on social knowledge. In *19th Int'l. Conf. on Information and Knowledge Management*, 189–198.

McCallum, A. K.; Nigam, K.; and Ungar, L. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. of the 6th International Conf. on Knowledge Discovery and Data Mining*, 169–178.

Michelson, M., and Knoblock, C. 2006. Learning blocking schemes for record linkage. In *Proc. of the 21st National Conf. on Artificial Intelligence - Vol. 1*, 440–445.

Ng, V., and Cardie, C. 2002. Improving machine learning approaches to coreference resolution. In *40th Annual Meeting on Association for Computational Linguistics*, 104–111.

Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. Y. 2007. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, 759–766. New York, NY, USA: ACM.

Rattenbury, T., and Naaman, M. 2009. Methods for extracting place semantics from flickr tags. *ACM Transactions on the Web* 3(1):1.

Rendle, S., and Schmidt-Thieme, L. 2006. Object identification with constraints. In *Proceedings of the 6th IEEE International Conference on Data Mining*, 1026–1031.

Rendle, S., and Schmidt-Thieme, L. 2008a. Active learning of equivalence relations by minimizing the expected loss using constraint inference. In *Proceedings of 8th IEEE International Conference on Data Mining*, 1001–1006.

Rendle, S., and Schmidt-Thieme, L. 2008b. Scaling record linkage to non-uniform distributed class sizes. *Advances in Knowledge Discovery and Data Mining* 5012:308–319.

Romano, L.; Buza, K.; Giuliano, C.; and Schmidt-Thieme, L. 2009. Xmedia: Web people search by clustering with machinely learned similarity measures. In *2nd Web People Search Evaluation Workshop, 18th WWW Conference*.

Sarawagi, S., and Bhamidipaty, A. 2002. Interactive deduplication using active learning. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 269–278.

Singla, P., and Domingos, P. 2005. Object identification with attribute-mediated dependences. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 297–308.

Soon, W.; Ng, H.; and Lim, D. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* 27(4):521–544.

Strehl, A., and Ghosh, J. 2003. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3:583–617.